MATH 366 Final Project

Taylor Larrechea
Colorado Mesa University

May 13, 2019

21.6 Methods for Parabolic PDEs.

The last two sections concerned elliptic PDEs, ad we now turn to parabolic PDEs. Recall that the definitions of elliptic, parabolic, and hyperbolic PDE's were given in Sec. 21.4 There it was also mentioned that the general behavior of solutions differs from type to type, and so do the problems of practical interest. This reflects on numerics as follows.

For all three types, one replaces the PDE by a corresponding difference equation, but for *parabolic* and *hperbolic* PDEs this does not automatically guarantee the **convergence** of the approximate solution to the exact solution as the mesh $h \to 0$; in fact, it does not even guarantee convergence at all. For all these two types of PDEs one needs additional conditions (inequalities) to assure convergence and **stability**, the latter meaning that small perturbations in the initial data (or small errors at any time) cause only small changes at later times.

In this section we explain the numeric solution of the prototype of parabolic PDEs, the one-dimensional heat equation

$$u_t = c^2 u_{xx}. \qquad \text{(c constant)}$$

This PDE is usually considered for x in some fixed interval, say $0 \leqq x \leqq L$, and time $t \geqq 0$, and one prescribes the initial temperature $u(x, 0) = f(x)$ (f given) and boundary conditions at $x = 0$ and $x = L$ for all $t \geqq 0$, for instance, $u(0, t) = 0$, $u(L, t) = 0$. We may assume $c = 1$ and $L = 1$; this can always be accomplished by a linear transformation of x and t (Prob. 1). Then the **heat equation** and those conditions are

$$u_t = u_{tt} \qquad 0 \leq x \leq L,\ t \geq 0 \qquad (21.1)$$

$$u(x, 0) = f(x) \qquad \text{(Initial condition)} \qquad (21.2)$$

$$u(0, t) = u(1, t) = 0 \qquad \text{(Boundary conditions)}. \qquad (21.3)$$

A simple finite difference approximation of (1) is [see (6a) in Sec. 21.4; j is the number of the ***time step***]

$$\frac{1}{k}(u_{i,j+1} - u_{ij}) = \frac{1}{h^2}(u_{i+1,j} - 2u_{ij} + u_{i-1,j}). \qquad (21.4)$$

Figure 465 shows a corresponding grid and mesh points. The mesh size is h in the x-direction and k and the t-direction. Formula (4) involves the four points shown in Fig. 466. On the left in (4) we have used a *forward* difference quotient since we have no information for negative t at the start. From (4) we calculate $u_{i,j+1}$, which corresponds to time row $j + 1$, in terms of the three other u that correspond to time row j. Solving (4) for $u_{i,j+1}$, we have

$$u_{i,j+1} = (1 - 2r)u_{ij} + r(u_{i+1,j} + u_{i-1,j}) \qquad r = \frac{k}{h^2}. \qquad (21.5)$$

Computations by this **explicit method** based on (5) are simple. However, if can be shown that crucial to the convergence of this method is the condition

$$r = \frac{k}{h^2} \leqq \frac{1}{2}. \tag{21.6}$$

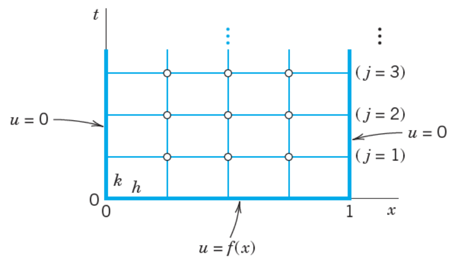


Fig. 465. Grid and mesh points corresponding to (4), (5)

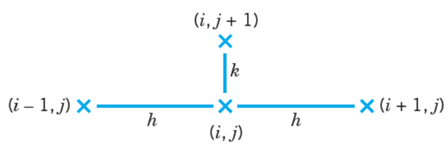


Fig. 466. The four points in (4) and (5)

That is, u_{ij} should have a positive coefficient in (5) or (for $r = \frac{1}{2}$) be absent from (5). Intuitively, (6) means that we should not move too fast in the t-direction. An example is given below.

Crank-Nicolson Method

Condition (6) is a handicap in practice. Indeed, to attain sufficient accuracy, we have to choose h small, which makes k very small by (6). For example, if $h = 0.1$, then $k \leqq 0.005$. Accordingly, we should look for a more satisfactory discretization of the heat equation.

A method that imposes no restriction on $r = k/h^2$ is the **Crank-Nicolson (CN) method** [1], which uses values of u at the six points in Fig. 467. The idea of

[1]JOHN CRANK (1916-2006), English mathematician and physicist at Courtaulds Fundamental Research Laboratory, professor at Brunel University, England. Student of Sir WILLIAM LAWRENCE BRAGG (1890-1971), Australian British physicist, who with his father, Fir WILLIAM HENRY BRAGG (1862-1942) won the Nobel Prize in physics in 1915 for their fundamental work in X-ray crystallography. (This is the only case where a father and a son shared the Nobel Prize for the same research. Furthermore, W.L, Bragg is the youngest Nobel laureate ever.) PHYLLIS NICOLSON (1917-1968), English mathematician, professor at the University of Leeds, England.

the method is the replacement of the difference quotient on the right side of (4) by $\frac{1}{2}$ times the sum of two such difference quotients at two time rows (see Fig. 467). Instead of (4) we then have

$$\frac{1}{k}(u_{i,j+1}-u_{ij}) = \frac{1}{2h^2}(u_{i+1,j}-2u_{ij}+u_{i-1,j}) + \frac{1}{2h^2}(u_{i+1,j+1}-2u_{i,j+1}+u_{i-1,j+1}). \tag{21.7}$$

Multiplying by $2k$ and writing $r = k/h^2$ as before, we collect the terms corresponding to time row $j+1$ on the left and the terms corresponding to time row j on the right:

$$(2+2r)u_{i,j+1} - r(u_{i+1,j+1}+u_{i-1,j+1}) = (2-2r)u_{ij} + r(u_{i+1,j}+u_{i-1,j}). \tag{21.8}$$

How do we use (8)? In general, the three values on the left are unknown, whereas the three values on the right are known. If we divide the x-interval $0 \leqq x \leqq 1$ in (1) into n equals intervals, we have $n-1$ internal mesh points per time row (see Fig. 465, where $n = 4$). Then for $j = 0$ and $i = 1, \dots, n-1$, formula (8) gives a linear system of $n-1$ equations for the $n-1$ unknown values $u_{11}, u_{21}, \dots, u_{n-1,1}$ in the first time row in terms of the initial values $u_{00}, u_{10}, \dots, u_{n0}$ and the boundary values $u_{01}(=0), u_{n1}(=0)$. Similarly for $j = 1, j = 2$, and so on; that is, for each time row we have to solve such a linear system of $n-1$ equations resulting from (8).

Although $r = k/h^2$ is no longer restricted, smaller r will still give better results. In practice, one chooses a k by which one can save a considerable amount of work, without making r too large. For instance, often a good choice is $r = 1$ (which would be impossible in the previous method). Then (8) becomes simply

$$4u_{i,j+1} - u_{i+1,j+1} - u_{i-1,j+1} = u_{i+1,j} + u_{i-1,j}. \tag{21.9}$$

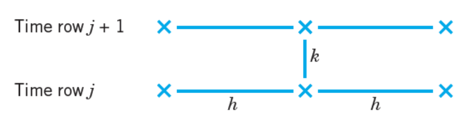


Fig. 467. The six points in the Crank-Nicolson formulas (7) and (8)

Example 1: Temperature in a Metal Bar. Crank-Nicolson Method, Explicit Method

Consider a laterally insulated metal bar of length 1 and such that $c^2 = 1$ in the heat equation. Suppose that the ends of the bar are kept at temperature

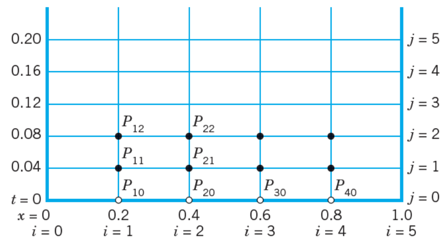


Fig. 468. Grid in Example 1

$u = 0^oC$ and the temperature in the bar at some instant — call it $t = 0$ — is $f(x) = \sin \pi x$. Applying the Crank-Nicolson method $h = 0.2$ and $r = 1$, find the temperature $u(x, t)$ in the bar for $0 \leqq t \leqq 0.2$. Compare the results with the exact solution. Also apply (5) with an r satisfying (6), say, $r = 0.25$, and with values not satisfying (6), say, $r = 1$ and $r = 2.5$. The initial conditions can be seen in the Figure below.

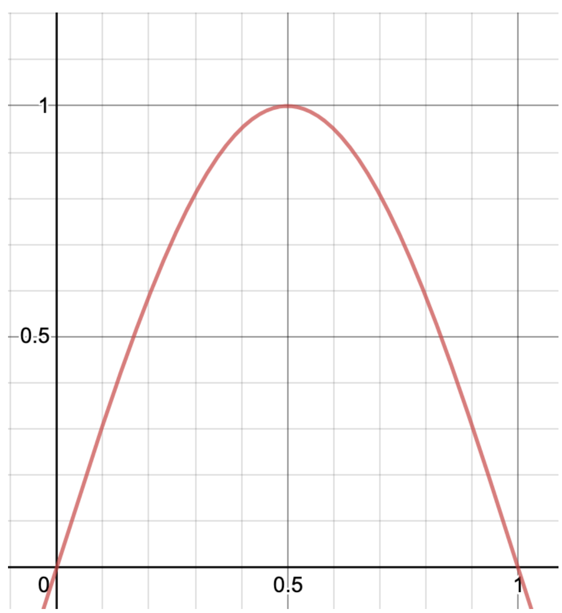


Fig. 468a Initial conditions of metal bar.

Solution by Crank-Nicolson. Since $r = 1$, formula (8) takes the form (9). Since $h = 0.2$ and $r = k/h^2 = 0.04$. Hence we have to do 5 steps. Figure 468 shows the grid. We shall need the initial values

$$u_{10} = \sin 0.2\pi = 0.587785, \qquad u_{20} = \sin 0.4\pi = 0.951057.$$

Also, $u_{30} = u_{20}$ and $u_{40} = u_{10}$. (Recall that u_{10} means u at P_{10} in Fig. 468, etc.) In each time row in Fig. 468 there are 4 internal mesh points. Hence in each time step we would have to solve 4 equations in 4 unknowns. But since the initial temperature distribution is symmetric with respect to $x = 0.5$, and $u = 0$ at both ends for all t, we have $u_{31} = u_{21}$, $u_{41} = u_{11}$ in the first time row and similarly for the other rows. This reduce each system to 2 equations in 2 unknowns. By (9), since $u_{31} = u_{21}$ and $u_{01} = 0$, for $j = 0$ these equations are

$$\begin{aligned} &(i=1) \qquad && 4u_{11} - u_{21} = u_{00} + u_{20} = 0.951057 \\ &(i=2) \qquad && -u_{11} + 4u_{21} - u_{21} = u_{10} + u_{20} = 1.538842. \end{aligned}$$

The solution is $u_{11} = 0.399274$, $u_{21} = 0.646039$. Similarly, for time row $j = 1$ we have the system

$$\begin{aligned} &(i=1) \qquad && 4u_{12} - u_{22} = u_{01} + u_{21} = 0.646039 \\ &(i=2) \qquad && -u_{12} + 3u_{22} = u_{11} + u_{21} = 1.045313. \end{aligned}$$

We now wish to show these system of equations in matrix form.

$$\begin{bmatrix} 4 & -1 \\ -1 & 3 \end{bmatrix} \begin{bmatrix} u_{11} \\ u_{21} \end{bmatrix} = \begin{bmatrix} 0.951 \\ 1.538 \end{bmatrix}$$

$$\begin{bmatrix} 4 & -1 \\ -1 & 3 \end{bmatrix} \begin{bmatrix} u_{12} \\ u_{22} \end{bmatrix} = \begin{bmatrix} 0.646 \\ 1.045 \end{bmatrix}$$

$$\begin{bmatrix} 4 & -1 \\ -1 & 3 \end{bmatrix} \begin{bmatrix} u_{13} \\ u_{23} \end{bmatrix} = \begin{bmatrix} 0.438 \\ 0.710 \end{bmatrix}$$

$$\begin{bmatrix} 4 & -1 \\ -1 & 3 \end{bmatrix} \begin{bmatrix} u_{14} \\ u_{24} \end{bmatrix} = \begin{bmatrix} 0.298 \\ 0.481 \end{bmatrix}$$

$$\begin{bmatrix} 4 & -1 \\ -1 & 3 \end{bmatrix} \begin{bmatrix} u_{15} \\ u_{25} \end{bmatrix} = \begin{bmatrix} 0.202 \\ 0.329 \end{bmatrix}$$

The solution is $u_{12} = 0.271221$, $u_{22} = 0.438844$, and so on. This process is repeated for all of the above matrices and the following results are given. It should be noted that $u_{11} = u_{41}$ and $u_{21} = u_{31}$ due to symmetry. This pattern repeats for all values of j. The code that was used to solve these matrices can be seen in the appendix titled, "MATLAB Appendix". This gives the temperature distribution (Fi.g 469):

t	$x=0$	$x=0.2$	$x=0.4$	$x=0.6$	$x=0.8$	$x=1$
0.00	0	0.588	0.951	0.951	0.588	0
0.04	0	0.399	0.646	0.646	0.399	0
0.08	0	0.271	0.439	0.439	0.271	0
0.12	0	0.184	0.298	0.298	0.184	0
0.16	0	0.125	0.202	0.202	0.125	0
0.20	0	0.085	0.138	0.138	0.085	0

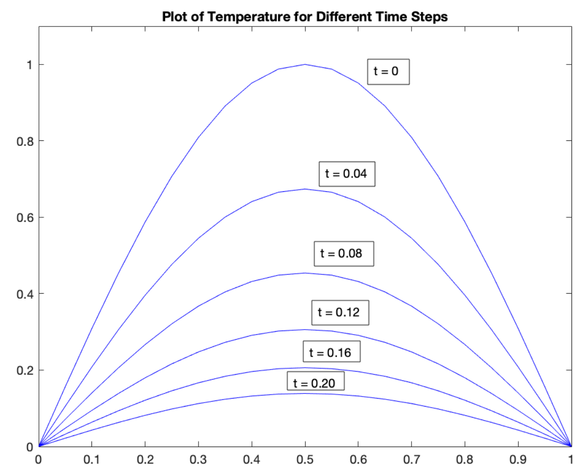


Fig. 469. Temperature distribution in the bar in Example 1. The script that was used to create this graph can be seen in the "MATLAB Appendix".

Comparison with the exact solution. The present problem can be solved exactly by separating variables (Sec. 12.5); the result is

$$u(x,t) = \sin \pi x \, e^{-\pi^2 t}. \tag{21.10}$$

Solution by the explicit method (5) with r=0.25. For $h = 0.2$ and $r = k/h^2 = 0.25$ we have $k = rh^2 = 0.25 \cdot 0.04 = 0.01$. Hence we have to perform 4 times as many steps as with the Crank-Nicolson method! Formula (5) with $r = 0.25$ is

$$u_{i,j+1} = 0.25(u_{i-1,j} + 2u_{ij} + u_{i+1,j}). \tag{21.11}$$

We can again make use of the symmetry. For $j = 0$ we need $u_{00} = 0$, $u_{10} = 0.587785$ (see p. 939), $u_{20} = u_{30} = 0.951057$ and compute

$$u_{11} = 0.25(u_{00} + 2u_{10} + u_{20}) = 0.531657$$
$$u_{21} = 0.25(u_{10} + 2u_{20} + u_{30}) = 0.25(u_{10} + 3u_{20}) = 0.860239.$$

Of course we can omit the boundary terms $u_{01} = 0$, $u_{02} = 0$, ...from the formulas. For $j = 1$ we compute

$$u_{12} = 0.25(2u_{11} + u_{21}) = 0.480888$$
$$u_{22} = 0.25(u_{11} + 3u_{21}) = 0.778094$$

and so on. We have to perform 20 steps instead of the 5 CN steps, but the numeric values show that the accuracy is only about the same as that of the Crank-Nicolson values CN. The exact 3D-values follow from (10).

t	x=0.2 CN	x=0.2 By (11)	x=0.2 Exact	x=0.4 CN	x=0.4 By (11)	x=0.4 Exact
0.04	0.399	0.393	0.396	0.646	0.637	0.641
0.08	0.271	0.263	0.267	0.439	0.426	0.432
0.12	0.184	0.176	0.180	0.298	0.285	0.291
0.16	0.125	0.118	0.121	0.202	0.191	0.196
0.20	0.085	0.079	0.082	0.138	0.128	0.132

Failure of (5) with r violating (6). Formula (5) with $h = 0.2$ and $r = 1$ — which violates (6) — is

$$u_{i,j+1} = u_{i-1,j} - u_{ij} + u_{i+1,j}$$

and gives very poor values; some of these are

t	*x=0.2*	Exact	*x=0.4*	Exact
0.04	0.363	0.396	0.588	0.641
0.12	0.139	0.180	0.225	0.291
0.20	0.053	0.082	0.086	0.132

Formula (5) with an even larger $r = 2.5$ (and $h = 0.2$ as before) gives completely nonsensical results; some of these are

t	*x=0.2*	Exact	*x=0.4*	Exact
0.1	0.0265	0.2191	0.0429	0.3545
0.3	0.0001	0.0304	0.0001	0.0492.

□

21.7 Methods for Hyperbolic PDEs.

In this section we consider the numeric solution of problems involving hyperbolic PDEs. We explain a standard method in terms of a typical setting for the prototype of a hyperbolic PDE, the **wave equation:**

$$u_{tt} = u_{xx} \qquad 0 \leqq x \leqq 1, t \geqq 0 \tag{22.1}$$

$$u(x,0) = f(x) \qquad 0 \leqq x \leqq 1, t \geqq 0 \tag{22.2}$$

$$u_t(x,0) = g(x) \qquad \text{(Given initial velocity)} \tag{22.3}$$

$$u(0,t) = u(1,t) = 0 \qquad \text{(Boundary conditions).} \tag{22.4}$$

Note that an equation $u_{tt} = c^2 u_{xx}$ and another x-interval can be reduced to the form (1) by a linear transformation of x and t. This is similar to Sec. 21.6, Prob. 1.

For instance, (1)—(4) is the model of a vibrating elastic string with fixed ends at $x = 0$ and $x = 1$ (see Sec. 12.2). Although an analytic solution of the problem is given in (13), Sec. 12.4, we use the problem for explaining basic ideas of the numeric approach that are also relevant for more complicated hyperbolic PDEs.

Replacing the derivatives by difference quotients as before, we obtain fro (1) [see (6) in Sec. 21.4 with $y = t$]

$$\frac{1}{k^2}(u_{i,j+1} - 2u_{ij} + u_{i,j-1}) = \frac{1}{h^2}(u_{i+1,j} - 2u_{ij} + u_{i-1,j}) \tag{22.5}$$

where h is the mesh size in x, and k is the mesh size in t. This difference equation relates 5 points as shown in Fig. 470a. It suggests a rectangular grid similar to the grids for parabolic equations in the preceding section. We choose $r^* = k^2/h^2 = 1$. Then u_{ij} drops out and we have

$$u_{i,j+1} = u_{i-1,j} + u_{i+1,j} - u_{i,j-1} \qquad \text{(Fig. 470b).} \tag{22.6}$$

It can be shown that for $0 < r^* \leqq 1$ the present **explicit method** is stable, so that from (6) we may expect reasonable results for initial data that have no discontinuities. (For a hyperbolic PDE the latter would propagate into the solution domain—a phenomenon that would be difficult to deal with on our present grid. For unconditionally stable **implicit methods** see [E1] in App. 1.)

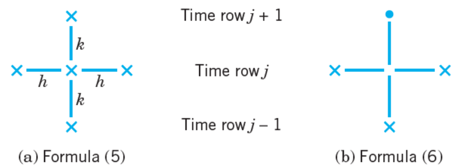


Fig. 470. Mesh points used in (5) and (6)

Equation (6) still involves 3 time steps $j-1$, j, $j+1$, whereas the formulas in the parabolic case involved only 2 time steps. Furthermore, we now have 2 initial conditions. So we ask how we get started and how we can use the initial condition (3). This can be done as follows.

From $u_t(x,0) = g(x)$ we derive the difference formula

$$\frac{1}{2k}(u_{i1} - u_{i,-1}) = g_i, \quad \text{hence} \quad u_{i,-1} = u_{i1} - 2kg_i \tag{22.7}$$

where $g_i = g(ih)$. For $t = 0$, that is $j = 0$, equation (6) is

$$u_{i1} = u_{i-1,0} + u_{i+1,0} - u_{i,-1}.$$

Into this we substitute $u_{i,-1}$ as given in (7). We obtain $u_{i1} = u_{i-1,0} + u_{i+1,0} - u_{i1} + 2kg_i$ and by simplification

$$u_{i1} = \frac{1}{2}(u_{i-1,0} + u_{i+1,0}) + kg_i. \tag{22.8}$$

This expresses u_{i1} in terms of the initial data. It is for the beginning only. Then use (6).

EXAMPLE 1: Vibrating String, Wave Equation

Apply the present method with $h = k = 0.2$ to the problem (1)—(4), where

$$f(x) = \sin \pi x, \quad g(x) = 0.$$

Solution. The grid is the same as in Fig. 468, Sec. 21.6, except for the values of t, which now are 0.2, 0.4, ... (instead of 0.04, 0.08, ...), The initial values of u_{00}, u_{10}, ... are the same as in Example 1, Sec. 21.6. From (8) and $g(x) = 0$ we have

$$u_{i1} = \frac{1}{2}(u_{i-1,0} + u_{i+1,0}).$$

From this we compute, using $u_{10} = u_{40} = \sin 0.2\pi = 0.587785$, $u_{20} = u_{30} = 0.951057$,

$$(i=1) \quad u_{11} = \frac{1}{2}(u_{00} + u_{20}) = \frac{1}{2} \cdot 0.951057 = 0.475528$$
$$(i=2) \quad u_{21} = \frac{1}{2}(u_{10} + u_{30}) = \frac{1}{2} \cdot 1.538842 = 0.769421$$

and $u_{31} = u_{21}$, $u_{41} = u_{11}$ by symmetry as in Sec. 21.6, Example 1. From (6) with $j = 1$ we now compute, using $u_{01} = u_{02} = \cdots = 0$,

$$(i=1) \quad u_{12} = u_{01} + u_{21} - u_{10} = 0.769421 - 0.587785 = 0.181636$$
$$(i=2) \quad u_{22} = u_{11} + u_{31} - u_{20} = 0.475528 + 0.769421 - 0.951057 = 0.293892,$$

and $u_{32} = u_{22}$, $u_{42} = u_{12}$ by symmetry; and so on. We thus obtain the following values of the displacement $u(x,t)$ of the string over the first half-cycle:

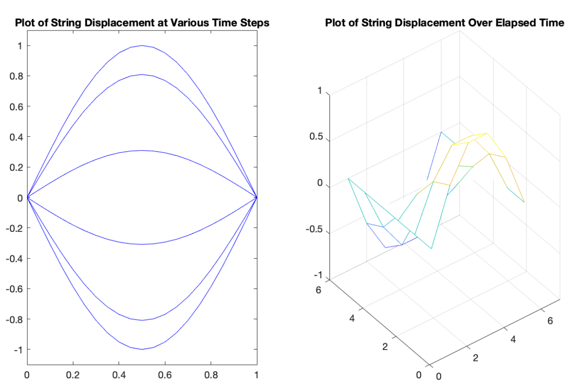
t	$x=0$	$x=0.2$	$x=0.4$	$x=0.6$	$x=0.8$	$x=1$
0.0	0	0.588	0.951	0.951	0.588	0
0.2	0	0.476	0.769	0.769	0.476	0
0.4	0	0.182	0.294	0.294	0.182	0
0.6	0	-0.182	-0.294	-0.294	-0.182	0
0.8	0	-0.476	-0.769	-0.769	-0.476	0
1.0	0	-0.588	-0.951	-0.951	-0.588	0

These values are exact to 3D (3 decimals), the exact solution of the problem being (see Sec. 12.3)

$$u(x,t) = \sin \pi x \cos \pi t.$$

The reason for the exactness follows from d'Alembert's solution (4), Sec. 12.4. (See Prob. 4, below.)

The following Figure is the exact solution for Example 1.



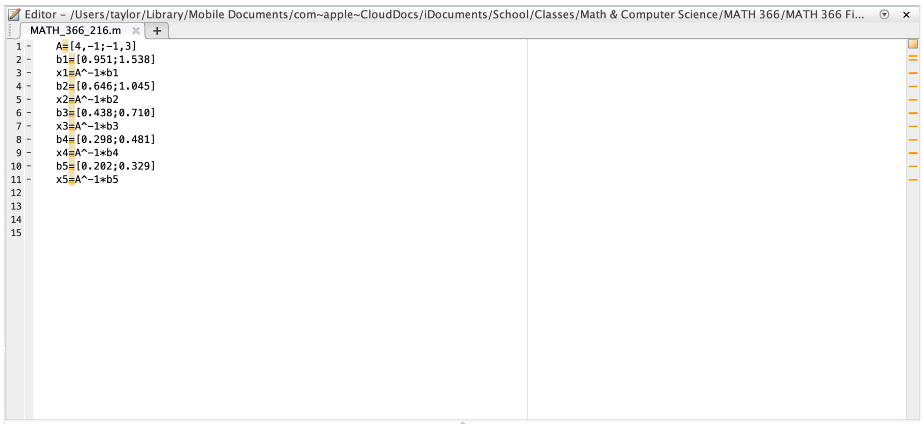
Exact solution for Example 1. The script for this program can once again be found in the, "MATLAB Appendix".

□

This is the end of Chap. 21 on numerics for ODEs and PDEs, a field that continues to develop rapidly in both applications and theoretical research. Much of the activity in the field is due to the computer serving as an invaluable tool for solving large-scale and complicated practical problems as well as for testing and experimenting with innovative ideas. These ideas could be small or major improvements on existing numeric algorithms or testing new algorithms as ell as other ideas.

MATLAB Appendix

Script to solve the matrices in 21.6:



Editor – /Users/taylor/Library/Mobile Documents/com~apple~CloudDocs/iDocuments/School/Classes/Math & Computer Science/MATH 366/MATH 366 Fi...

MATH_366_216.m

```
A=[4,-1;-1,3]
b1=[0.951;1.538]
x1=A^-1*b1
b2=[0.646;1.045]
x2=A^-1*b2
b3=[0.438;0.710]
x3=A^-1*b3
b4=[0.298;0.481]
x4=A^-1*b4
b5=[0.202;0.329]
x5=A^-1*b5
```

MATLAB script used to solve the five matrices from Ch. 21.6.

The output for the first set of matrices is:

Command Window

```
>> MATH_366_216

A =

     4    -1
    -1     3


b1 =

        0.951
        1.538


x1 =

      0.39918
      0.64573


b2 =

        0.646
        1.045


x2 =

      0.27118
      0.43873
```

MATLAB output for the first set of matrices found in Ch. 21.6.

The output for the second set of matrices is:

```
b3 =

        0.438
         0.71


x3 =

        0.184
        0.298


b4 =

        0.298
        0.481


x4 =

        0.125
        0.202


b5 =

        0.202
        0.329


x5 =

        0.085
        0.138
```

MATLAB output for the second set of matrices found in Ch. 21.6.

The script that was used to create Figure 469 can be seen below.

Editor – /Users/Tate/Library/Mobile Documents/com~apple~CloudDocs/iDocuments/School/Classes/Math & Computer Science/MATH 3...

MATH_366_Fig469.m

```
%The following commands graph the analytical solution u(x,t) %for different time values.
%The analytical solution is u(x,t) = sin(pi*x)*exp(-pi^2*t).%The analytical solution is sampled on x = 20 spatial nodes.
M = 20;
x = [0:1/M:1]';
f1 = sin(pi*x)*exp(-pi^2*0);
f2 = sin(pi*x)*exp(-pi^2*0.04);
f3 = sin(pi*x)*exp(-pi^2*0.08);
f4 = sin(pi*x)*exp(-pi^2*0.12);
f5 = sin(pi*x)*exp(-pi^2*0.16);
f6 = sin(pi*x)*exp(-pi^2*0.20);
%Plot analytical solution and numerical solution.
plot(x,[f1,f2,f3,f4,f5,f6],'b')
title('Plot of Temperature for Different Time Steps')
axis([0 1 0 1.1])
```

MATLAB script used to create Figure 469.

The script for Example 1 in Ch. 21.7.

Editor – /Users/Tate/Library/Mobile Documents/com~apple~CloudDocs/iDocuments/School/Classes/Math & Computer Science/MATH 3...

MATH_366_217.m

```
% This script implements a numerical method for solving the
% hyperbolic PDE as shown in Example 1 of Ch. 21.7 of our book.
% The book uses m = 2 internal nodes & symmetry per time step

n = 6;  %Number of time steps from t = 0 to t = 1.
N = 7;  %Number of columns in solution matrix S

h = 0.2;    %Mesh size for x
k = h;  %Mesh size for t
xnodes = [0:h:1];
tnodes = [0:k:1];
f = sin(pi*xnodes);     %Initial spatial distribution

% Construct solution matrix S
S = zeros(n,N);
S(:,1) =  tnodes;   %This is to match first columnof table in book
S(1,2:N) = f;       %First row of S is f

% The second row uses Eqn (8) from book (initial conditions using g)
for i = 3:N-1
    S(2,i)=0.5*(S(1,i-1)+S(1,i+1))
end
```

MATLAB Script for Example 1 in Ch. 21.7 (1 of 3)

```
% To compute subsequent rows of S, we use Eqn (6) from our book.
% The book uses row symmetry to reduce flops. However, the loop below
% is a direct application of Eqn (6) and does not use row symmetry,
% which is OK for us.
for k = 3:n
    for i = 3:N-1
        S(k,i)=S(k-1,i-1)+S(k-1,i+1)-S(k-2,i)
    end
end
%S

% We next graph analytical solution u(x,t) for different time values.
% The analytical solution is u(x,t) = sin(pi*x)*cos(pi*t).
% The analytical solution will be sampled on x = 20 spatial nodes.

M = 20;
x = [0:1/M:1]';
f1 = sin(pi*x)*cos(pi*0);
f2 = sin(pi*x)*cos(pi*0.2);
f3 = sin(pi*x)*cos(pi*0.4);
f4 = sin(pi*x)*cos(pi*0.6);
f5 = sin(pi*x)*cos(pi*0.8);
f6 = sin(pi*x)*cos(pi*1.0);
```

MATLAB Script for Example 1 in Ch. 21.7 (2 of 3)

```
% Plot analytical solution and numerical solution.
figure

subplot(1,2,1)
plot(x,[f1,f2,f3,f4,f5,f6],'b')
title('Plot of String Displacement at Various Time Steps')
axis([0 1 -1.1 1.1])

subplot(1,2,2)
mesh(S(:,2:N))
title('Plot of String Displacement Over Elapsed Time')
axis([0 N 0 n -1 1])
```

MATLAB Script for Example 1 in Ch. 21.7 (3 of 3)

MATLAB output for Example 1 in Ch. 21.7.

```
S =

         0         0    0.5878    0.9511    0.9511    0.5878    0.0000
    0.2000         0    0.4755    0.7694    0.7694    0.4755         0
    0.4000         0    0.1816    0.2939    0.2939    0.1816         0
    0.6000         0   -0.1816   -0.2939   -0.2939   -0.1816         0
    0.8000         0   -0.4755   -0.7694   -0.7694   -0.4755         0
    1.0000         0   -0.5878   -0.9511   -0.9511   -0.5878         0
```

MATLAB output for Example 1 in Ch. 21.7.